



Signature Gateway Profile of the OASIS Digital Signature Service

Committee Draft, 13 June 2005

Document identifier:

dss-v1.0-spec-cd-SignatureGatewayProfile-r01

Location:

<http://www.oasis-open.org/committees/dss>

Editor:

Glenn Benson, JPMorgan <glenn.benson@jpmorgan.com>

Contributors:

Burt Kaliski, RSA Security <BKaliski@rsasecurity.com>

John Linn, RSA Security <jlinn@rsasecurity.com>

Trevor Perrin, Individual <trevp@trevp.net>

Abstract:

This draft profiles the OASIS DSS core protocol for signature gateway transformation processing. This profile is intended to be generic, so it may be combined with other profiles freely.

Status:

This is a **Committee Draft** produced by the OASIS Digital Signature Service Technical Committee. Committee members should send comments on this draft to dss@lists.oasis-open.org.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Digital Signature Service TC web page at <http://www.oasis-open.org/committees/dss/ipr.php>.

27 Table of Contents

28	1	Introduction	3
29	1.1	Profile Type	3
30	1.2	Overview (Non-Normative).....	3
31	1.3	Request-Response Deployment Model	4
32	1.4	In-Line Deployment Model	4
33	1.5	Notation	5
34	1.6	Namespaces	6
35	2	Profile Features.....	7
36	2.1	Identifier.....	7
37	2.1.1	Core HTTP Transport Binding	7
38	2.1.2	Core SOAP 1.2 Transport Binding.....	7
39	2.1.3	Other Transport Bindings Defined as Concrete Sub-Profiles	7
40	2.2	Scope	7
41	2.3	Relationship To Other Profiles	7
42	2.4	Signature Object.....	8
43	2.5	Transport Binding	8
44	2.6	Security Binding	8
45	3	Profile of Signing Protocol.....	9
46	3.1	Element <SignRequest>	9
47	3.2	Element <SignResponse>	9
48	4	Profile of Verifying Protocol.....	10
49	4.1	Element VerifyRequest	10
50	4.2	Element OptionalInputs	10
51	4.3	Element <VerifyResponse>	11
52	4.3.1	Element <ResultMajor>	11
53	4.3.2	Element <ResultMinor>	11
54	4.3.3	Element <OptionalOutputs>	11
55	5	Profile of Signatures.....	13
56	6	Server Processing Rules	14
57	6.1	VerifyRequest.....	14
58	7	Editorial Issues.....	Error! Bookmark not defined.
59	8	References.....	15
60	8.1	Normative	15
61		Appendix A. Revision History	16
62		Appendix B. Notices	17
63			

64 1 Introduction

65 1.1 Profile Type

66 An OASIS DSS profile has exactly one class: *concrete* or *abstract*. The most significant
67 difference between the two classes is that one may directly implement a concrete protocol;
68 however, one may not claim conformance of a specific realization to an abstract protocol. A
69 concrete profile sufficiently constrains the flexibility of the DSS core protocol [**DSSCore**] so that a
70 profile-compliant client and server should be interoperable at the levels of the protocol as defined
71 in the profile. An abstract profile requires further definition of a subordinate concrete profile
72 before an implementer may create a conformant realization.

73 This document identifies one abstract profile and two concrete profiles. The abstract profile
74 defines all definitions required for DSS interoperability with one exception: transmission binding.

75 The concrete profiles fill the gap by permitting an implementer to build a realization and claim
76 Signature Gateway Profile realization by both conforming to the abstract profile, and conforming
77 to a permissible transmission binding as defined in one of the concrete profiles.

78 The two concrete profiles identified in this document each a specific transmission binding:

- 79 • HTTP POST Transport Binding, or
- 80 • SOAP 1.2 Transport Binding.

81 The addition of security to these bindings is optional.

82 Subsequent revisions may either add new concrete profiles in separate documents, or as
83 modifications to this document.

84 The following sections describe how to understand the rest of this document.

85 1.2 Overview (Non-Normative)

86 This document standardizes a Signature Gateway by profiling the DSS signing and verifying
87 protocols [**DSSCore**]. This Signature Gateway transforms both *signing technology* and *credential*
88 *logistics*. The signing technology specifies the mechanisms through which one creates and
89 verifies a signature. Example technologies include, but are not limited to photocopied signatures,
90 Public Key Infrastructure signatures, and signatures defined using symmetric keying material (see
91 [**XMLDSIG**] for some symmetric specifications). Credential logistics, describes the means to
92 distribute credentials to remote parties; and the associated vehicle for distributing trust. Although
93 electronic means allows communication at a distance, geographic separation increases the
94 difficulty of trusting one's peers. Credentials overcome many of the geographic impediments to
95 trust; and the associated logistics securely define the means of managing the credential lifecycle,
96 e.g., distribution, revocation, renewal, and retirement.

97 Each kind of technology and logistics has its own distinct advantages and disadvantages. As a
98 result, no universal best-of-breed solution exists for all deployment scenarios. Some scenarios
99 require different solutions for distinct spaces; and a gateway serves as an intermediary
100 connector. The DSS Signature Gateway operates in the following use case. A signer applies its
101 signing credential to create a signature. The signer does not transmit the signature directly to a
102 recipient, because the recipient might not understand the signer's signature technology; and the
103 recipient may not trust the signer's credential. Instead, the signer sends the signature to a
104 mutually trusted Signature Gateway which transforms the signature into a format that the

105 recipient validates. The Gateway's transformation operation first validates the original signature,
106 and then creates a new signature. Consider the following example. An organization may allow
107 its employees and machines to trust communication that originates from within the security
108 perimeter, while requiring extra security for externally-originated messages. Rather than
109 distribute the means for secure interoperability throughout the enterprise and extranet, the
110 organization may establish a trusted Signature Gateway. The Gateway validates its incoming
111 messages from the external parties; and then marks the Gateway's stamp of approval which
112 downstream servers consume.

113 The signature gateway profile may operate in multiple different deployment models. Two
114 example models are described below.

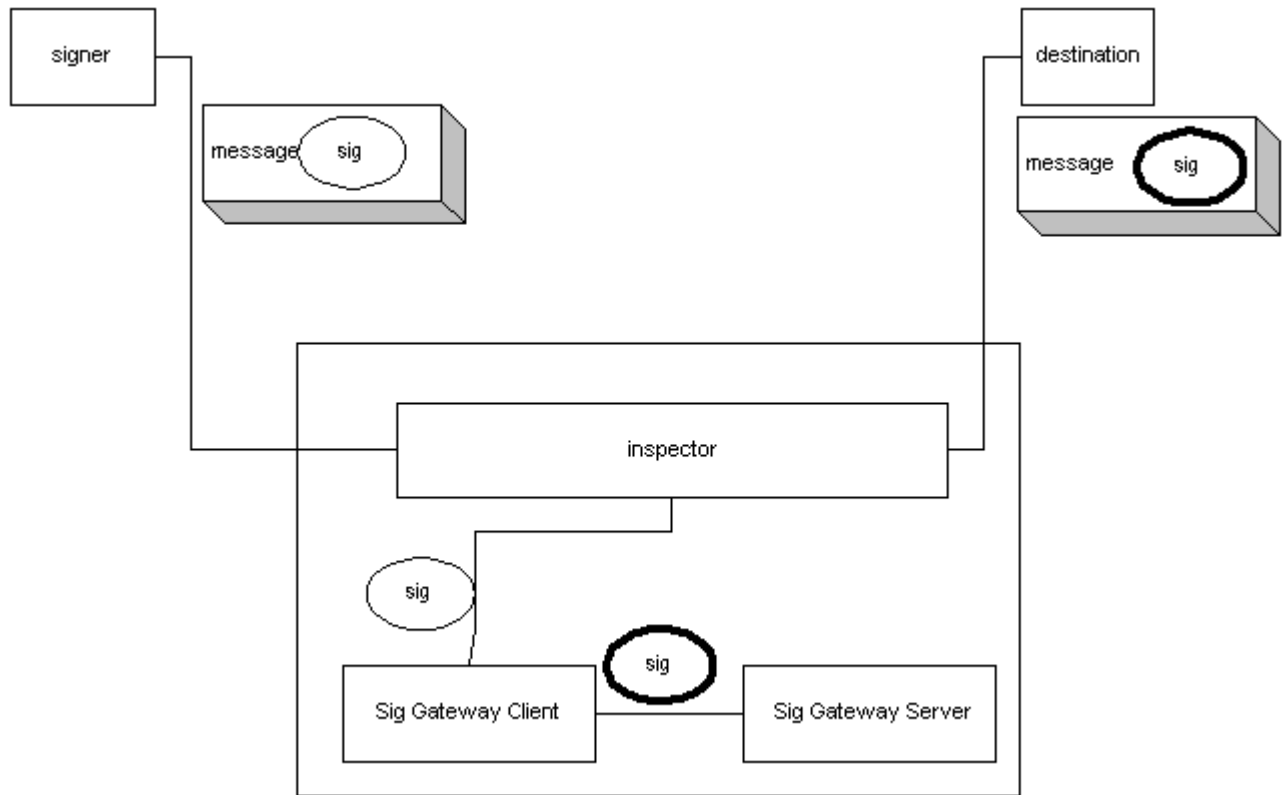
115 **1.3 Request-Response Deployment Model**

116 The request-response deployment model has three actors: signature client, DSS client, and DSS
117 Signature Gateway Server.

- 118 1. The signature client signs a document or transaction, and sends the signed data to the
119 DSS client.
- 120 2. The DSS client wraps the signed data in the context of DSS Signature Gateway Profile
121 VerifyRequest, and sends the request to the DSS Signature Gateway Server.
- 122 3. The DSS Signature Gateway server performs the necessary validation services, and
123 returns a DSS Signature Gateway VerifyResponse to the DSS client.

124 **1.4 In-Line Deployment Model**

125 Devices located at the security perimeter may combine Signature Gateway with other security
126 services. Consider for example, deep packet inspection firewalls, content-inspecting load
127 balancers, intelligent reverse proxies, or XML firewalls. These devices contain the technology to
128 inspect incoming communication while searching for signatures. When the device identifies a
129 signature within the context of a message, the device applies the Signature Gateway
130 transformation, and then forwards the modified communication to the destination. The Figure
131 below illustrates the constituent components:



132
133

134 The request-response deployment model has three actors: signer, inline proxy, and destination.
135 The inline proxy has three constituent components: inspector, Signature Gateway Client, and
136 Signature Gateway Server.

- 137 1. The signer sends a message that contains a signature to the in-line proxy.
138 2. The inspector component of the in-line proxy captures the message and searches for
139 signed data. If the inspector identifies signed data, then the inspector passes the signed
140 data to the DSS Signature Gateway Client.
141 3. The DSS Signature Gateway Client creates DSS Signature Gateway VerifyRequest using
142 the signed data. The DSS client sends this VerifyRequest to the DSS Signature Gateway
143 Server component.
144 4. The DSS Signature Gateway Server responds issuing a VerifyResponse.
145 5. The DSS client passes the response to the inspector component.
146 6. The inspector modifies the message per the response returned from the DSS Signature
147 Gateway Server and sends the modified message to a downstream, destination
148 application.

149 1.5 Notation

150 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
151 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be
152 interpreted as described in IETF RFC 2119 [RFC 2119]. These keywords are capitalized when
153 used to unambiguously specify requirements over protocol features and behavior that affect the

154 interoperability and security of implementations. When these words are not capitalized, they are
155 meant in their natural-language sense.

156 This specification uses the following typographical conventions in text: `<ns:Element>`,
157 `Attribute`, **Datatype**, `OtherCode`.

158 **1.6 Namespaces**

159 Conventional XML namespace prefixes are used in this document:

160 - The prefix `dss:` (or no prefix) stands for the DSS core namespace **[Core-XSD]**.

161 - The prefix `ds:` stands for the W3C XML Signature namespace **[XMLDSIG]**.

162 Applications MAY use different namespace prefixes, and MAY use whatever namespace
163 defaulting/scoping conventions they desire, as long as they are compliant with the Namespaces
164 in XML specification **[XML-ns]**.

165 2 Profile Features

166 2.1 Identifier

167 urn:asis:names:tc:dss:1.0:profiles:siggy

168 This identifier names an abstract profile. An <AdditionalProfile> identifier is mandatory in order to
169 name a subordinate concrete profile.

170 2.1.1 Core HTTP Transport Binding

171 The following <AdditionalProfile> specifies a concrete profile:

172 urn:asis:names:tc:dss:1.0:HTTP-POST-Transport-binding

173

174 This concrete profile requires:

- 175 - ingress: HTTP POST Transport binding as specified in the 1.0 core
- 176 - egress: unspecified

177

178 2.1.2 Core SOAP 1.2 Transport Binding

179 The following <AdditionalProfile> specifies a concrete profile:

180

181 urn:asis:names:tc:dss:1.0:SOAP-Transport-binding

182

183 This concrete profile requires:

- 184 - ingress: SOAP 1.2 Transport binding as specified in the 1.0 core
- 185 - egress: unspecified

186 2.1.3 Other Transport Bindings Defined as Concrete Sub-Profiles

187 If the transport binding is defined as in a subordinate profile, then add the requisite identifier as an
188 <AdditionalProfile>.

189

190 2.2 Scope

191 This document profiles the DSS signing and verifying protocols defined in **[DSSCore]** and profiles
192 XML signature format for a signature gateway. This document permits other signature formats
193 such as CMS **[RFC3369]**.

194 2.3 Relationship To Other Profiles

195 This profile is based directly on the **[DSSCore]**.

196

197 This document contains an abstract profile and two concrete protocols.

198 **2.4 Signature Object**

199 This profile supports the verification of incoming signatures and the production of a resultant
200 signature by the gateway. The profile **MUST** support XMLDSIG [**XMLDSIG**] for both incoming
201 and produced signatures. Other formats are optional. This means that a Signature Gateway
202 **MAY** accept incoming signatures in a non-XMLDSIG compliant format, e.g., CMS [**RFC3369**].

203 **2.5 Transport Binding**

204 The combination of this abstract profile and a permissible transport binding provides sufficient
205 specification for interoperability. For the transport bindings see the concrete protocols:
206 [**DSSCore**] HTTP POST Transport binding as named by urn:oasis:names:tc:dss:1.0:HTTP-
207 POST-Transport-binding, and [**DSSCore**] SOAP Transport Binding as named by
208 urn:oasis:names:tc:dss:1.0:SOAP-Transport-binding.

209 Other permissible transport bindings may be defined in subordinate concrete profiles.

210 **2.6 Security Binding**

211 A security binding is permissible but not required. If used, this profile does not specify or
212 constrain the security binding.

213 **3 Profile of Signing Protocol**

214 **3.1 Element <SignRequest>**

215 The <dss:SignRequest> is not supported in the Signature Gateway Profile.

216 **3.2 Element <SignResponse>**

217 The <dss:SignResponse> is not supported in the Signature Gateway Profile.

218 4 Profile of Verifying Protocol

219 4.1 Element VerifyRequest

220 4.2 Element OptionalInputs

221 The Signature Gateway Profile MAY support any client or server optional input defined in
222 **[DSSCore]**. However, some optional inputs are mandatory, or further clarified as described
223 below.

224 4.2.1.1 Optional input < ServicePolicy >

225 The Signature Gateway MUST support the optional input defined in **[DSSCore]**
226 `<dss:ServicePolicy>`. The `<dss:ServicePolicy>` MUST include a description of the
227 signature that the Signature Gateway accepts (ingress). In addition `<dss:ServicePolicy>`
228 MUST either include a description of the signature that the Signature Gateway produces (egress),
229 or explicitly note the policy for the egress signature using the term “unspecified”.

230

231 The `<dss:ServicePolicy>` specification for the ingress signature MUST include the following
232 items:

- 233 • The type of employed signature: **[XMLDSIG]** or **[RFC3369]**.
- 234 • Signature algorithm

235 The `<dss:ServicePolicy>` specification MAY include additional items such as signature
236 attributes, properties, or policies. Topics include, but are not limited to the items on the following
237 list:

- 238 • *Signed References and Properties*: Policy that determines if all the Signature Gateway
239 validates some, or all of the signed references and properties such as the manifest, and
240 timestamp.
- 241 • *Revocation*: Policy that specifies the rules by which the Signature Gateway checks
242 revocation on the input signature
- 243 • *Signature Coverage*: Policy that determines if the Gateway’s signature covers the
244 original document, the signature, the manifest, the signature properties, or some
245 combination of the above.
- 246 • *Timestamp*: Policy that specifies any requirement for a timestamp, including the format.
- 247 • *Revocation*: Policy that specifies the format, and server that provides revocation
248 information.

250 A Signature Gateway server MUST support at least one Service Policy. In the Signature
251 Gateway Profile, the `<dss:ServicePolicy>` is NOT optional, i.e., the client must provide it in
252 each request. A Signature Gateway MAY publish its service policy, where the means for
253 publication is outside the scope of DSS.

254 **4.2.1.2 OptionalInput < ReturnUpdatedSignature >**

255 Each <dss:VerifyRequest> MUST contain the optional input defined in [DSSCore]
256 <dss:ReturnUpdatedSignature>. The DSS Server MUST NOT sign the input document
257 unless it first validates the input <dss:SignatureObject> successfully.

258 **4.3 Element <VerifyResponse>**

259 **4.3.1 Element <ResultMajor>**

260 If the <dss:VerifyRequest> misses any of the required <dss:OptionalInputs>, then the
261 DSS server MUST return the following response in <dss:ResultMajor>.

262 urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError

263 **4.3.2 Element <ResultMinor>**

264

265 If the <dss:VerifyRequest> misses any of the required <dss:OptionalInputs>, then the
266 DSS server MUST return the following response in <dss:ResultMinor>:

267 urn:oasis:names:tc:dss:1.0:resultminor:siggy:NotSupported

268

269 The <dss:ResultMessage> SHOULD contain the identity of the missing
270 required <dss:OptionalInputs>.

271 **4.3.2.1 Signature type mismatch with requested key**

272 If the <dss:VerifyRequest> explicitly specifies a <dss:KeySelector>, where the Signature
273 Gateway's key is not valid, then the Signature Gateway MUST return an error with the following
274 code in <dss:ResultMinor>:

275

276 urn:oasis:names:tc:dss:1.0:resultminor:siggy:KeyNotSupported

277 **4.3.2.2 Signature policy not supported**

278 If the <dss:VerifyRequest> explicitly specifies an unsupported <dss:ServicePolicy>,
279 then the Signature Gateway MUST return an error with the following code in
280 <dss:ResultMinor>.

281

282 urn:oasis:names:tc:dss:1.0:resultminor:siggy:ServicePolicyNotSupported

283

284 **4.3.3 Element <OptionalOutputs>**

285 **4.3.3.1 OptionalOutput < UpdatedSignature >**

286 If the Signature Gateway Server fails to validate the signature in the VerifyRequest, then the
287 Signature Gateway Server MUST NOT include the <dss:UpdatedSignature>. If the Signature

288 Gateway Server successfully validates the signature in the VerifyRequest, then the Signature
289 Gateway Server SHOULD include the <dss:UpdatedSignature>

290 **5 Profile of Signatures**

291 The profile MAY support the XML Signature as defined in **[XMLDSIG]** or **[XAdES]**, within the
292 `<ds:object>` element of the XML signature.

293

294 The profile MAY support the CMS signature as defined in **[RFC3369]** specified as a
295 `<Base64Signature>` as defined in **[DSSCore]**.

296

297 **6 Server Processing Rules**

298 **6.1 VerifyRequest**

299 In addition to the processing specified in **[DSSCore]**, the DSS server additionally validates the
300 existence of all required optional inputs. The DSS server **MUST NOT** produce a signature unless
301 it first successfully validates the client's signature in accordance with the Service Policy.

302

303

304

305 **7 References**

306 **7.1 Normative**

- 307 **[Core-XSD]** T. Perrin et al. *DSS Schema*. OASIS, **(MONTH/YEAR TBD)**
- 308 **[DSSCore]** T. Perrin et al. *Digital Signature Service Core Protocols and Elements*. OASIS,
309 **(MONTH/YEAR TBD)**
- 310 **[DSS-XAdES]** Juan Carlos Cruellas et al. XAdES Profile of the OASIS Digital Signature Service
- 311 **[RFC 2119]** S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. IETF
312 RFC 2396, August 1998.
313 <http://www.ietf.org/rfc/rfc2396.txt>.
- 314 **[RFC3369]** R. Housley. *Cryptographic Message Syntax*. IETF RFC 3369, August 2002.
315 <http://www.ietf.org/rfc/rfc2459.txt>.
- 316 **[XAdES]** XML Advanced Electronic Signatures ETSI TS 101 903, February 2002 (*shortly*
317 *to be re-issued*)
318 http://pda.etsi.org/pda/home.asp?wiki_id=1UFEyx7ORuBCDGED3liJH
- 319 **[XML-ns]** T. Bray, D. Hollander, A. Layman. *Namespaces in XML*. W3C
320 Recommendation, January 1999.
321 <http://www.w3.org/TR/1999/REC-xml-names-19990114>
- 322 **[XMLDSIG]** D. Eastlake et al. *XML-Signature Syntax and Processing*. W3C
323 Recommendation, February 2002.
324 <http://www.w3.org/TR/1999/REC-xml-names-19990114>
- 325
- 326
- 327
- 328
- 329

Appendix A. Revision History

Rev	Date	By Whom	What
siggty-03	2004-13-Nov	Glenn Benson	Initial version with contributions from Burt Kaliski and John Linn
Siggty-06	2004-30-Dec	Glenn Benson	Update ServicePolicy per Trevor Perrin's suggestions; added to introduction; general cleanup
Siggty-07	2005-5-Mar	Glenn Benson	Converted from abstract to concrete profile in order to remove the transport binding
Siggty-08	2005-29-Mar	Glenn Benson	<ul style="list-style-type: none"> - single document with one abstract and two concrete identifiers: - Identifier only references the major version number - Introductory comments explaining additional concrete profiles may be made by either extending current document, or adding new documents
Siggty-09	2005-7-May	Glenn Benson	Incorporated comments from Nick Pope <ul style="list-style-type: none"> - added 'unspecified' egress policy - added support for CMS - cleaned up definitions of concrete extensions
Siggty-10	2005-19-May	Glenn Benson	Additional comments from Nick Pope: all updates to 4.2.1.1 <ul style="list-style-type: none"> - describe mandatory elements of ingress signature - overview optional elements of ingress and egress signature - simplify description of publication of service policy
cd-01	2005-13-June	Glenn Benson	Change status to committee draft

Appendix B. Notices

332 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
333 that might be claimed to pertain to the implementation or use of the technology described in this
334 document or the extent to which any license under such rights might or might not be available;
335 neither does it represent that it has made any effort to identify any such rights. Information on
336 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
337 website. Copies of claims of rights made available for publication and any assurances of licenses
338 to be made available, or the result of an attempt made to obtain a general license or permission
339 for the use of such proprietary rights by implementors or users of this specification, can be
340 obtained from the OASIS Executive Director.

341 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
342 applications, or other proprietary rights which may cover technology that may be required to
343 implement this specification. Please address the information to the OASIS Executive Director.

344 Copyright © OASIS Open 2003. *All Rights Reserved.*

345 This document and translations of it may be copied and furnished to others, and derivative works
346 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
347 published and distributed, in whole or in part, without restriction of any kind, provided that the
348 above copyright notice and this paragraph are included on all such copies and derivative works.
349 However, this document itself does not be modified in any way, such as by removing the
350 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS
351 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
352 Property Rights document must be followed, or as required to translate it into languages other
353 than English.

354 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
355 successors or assigns.

356 This document and the information contained herein is provided on an "AS IS" basis and OASIS
357 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
358 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
359 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
360 PARTICULAR PURPOSE.